BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Perusahaan

Tinjauan perusahaan akan mengetengahkan tentang sejarah, tempat dan kedudukan, dan visi dan misi, struktur organisasi dan deskripsi jabatan pada di PT. LASTERSIA PLASTISINDO Karawang.

2.1.1 Sejarah

Jawa barat merupakan salah satu daerah di indonesia yang kawasan industrinya paling pesat dan maju, dari mulai perusahaan yang bergerak di bidang Industri, Perdagangan, Jasa dan Lain sebagainya.

PT. Lastersia Plastisindo adalah salah satu perusahaan yang bergerak di bidang industri atau pengolahan daur ulang plastik yang menurut beberapa narasumber di media-media, bahwa limbah-limbah plastik didunia ini yang semakin tahun semakin melonjak, dan faktor-faktor negative pun bermunculan akibat limbah plastik yang berlebihan. Disinilah PT. Lastersia Plastisindo menjadi salah satu perusahaan yang solid, karena mengurangi limbah menjadi sesuatu yang berguna. PT yang bergerak dibidang pengolahan daur ulang plastik atau ”Plastic Recycle Industry” ini, didirikan tahun 1997 dan disahkan oleh menteri kehakiman dan hak asasi manusia menjadi sebuah perusahaan pada tanggal 13-April-2005 yang bertempat di Jln. Syech Quro.RT/RW 03/11 Plawad Kab.Karawang.

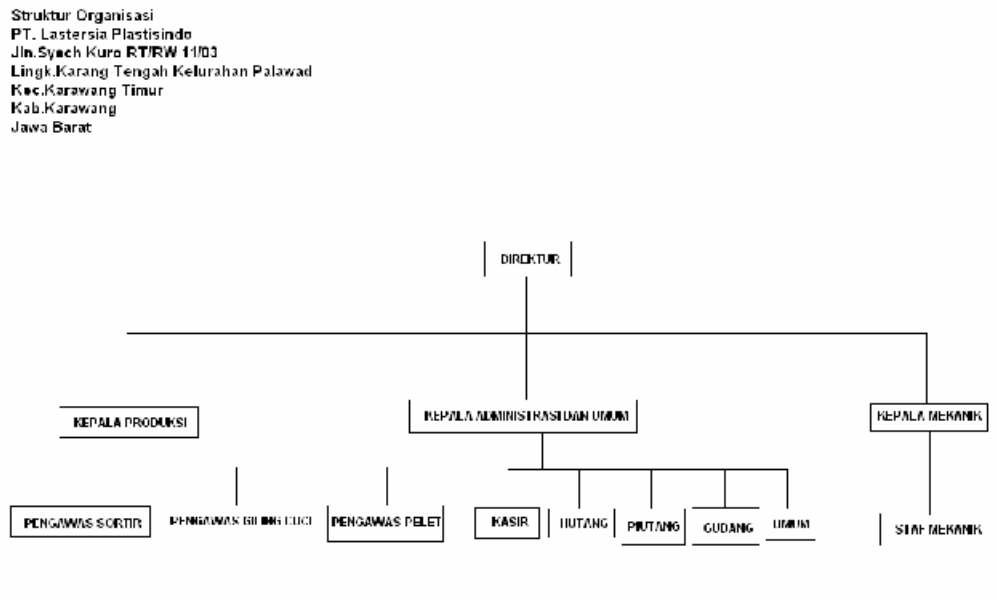
2.1.2 Tempat dan Kedudukan

PT. Lastersia Plastisindo Karawang berada di bawah naungan PT. Lastersia Plastisindo Kantor Cabang Utama Surabaya, yang bertempat di Jl. Krajan No. 01 Kemirisewu-Pandaan – Pasuruan – Jawa timur.

Kedudukan PT.Lastersia Plastisindo Surabaya adalah sebagai pensuplay barang mentahan untuk di olah di PT.Lastersia Plastisindo Karawang.

2.1.3 Struktur Organisasi

Struktur Organisasi di PT. Lastersia Plastisindo Karawang dapat dilihat pada gambar berikut ini :



Gambar 2.1 Struktur Organisasi PT.Lastersia Plastisindo Karawang

2.1.4 Deskripsi Jabatan

1. Direktur
    a. Merencanakan, mengorganisasikan dan mengendalikan seluruh kegiatan PT sesuai dengan rencana yang telah ditetapkan.

b. Memberikan pengarahan dan pembinaan kepada seluruh staff yang menjadi tanggung jawabnya.
c. Memberikan persetujuan dan pengesahan atas transaksi pemasukan dan pengeluaran.
d. Melakukan ikatan perjanjian kerja sama dengan pihak lain.
e. Melakukan evaluasi terhadap hasil kegiatan PT yang telah dicapai.
f. Mencari peluang-peluang usaha baru.

2. Kepala Produksi
   a. Melaporkan semua kegiatan usaha PT kepada direktur.
   b. Memproduksi barang hasil dari PT .
   c. Membantu dalam proses koordinasi dengan PT. Lastersia Plastisindo Surabaya.
   d. Mencari peluang-peluang untuk usaha baru.
3. Kepala Administrasi Dan Umum
   a. Merencanakan dan mengkoordinasi fungsi-fungsi keuangan.
   b. Menyiapkan dan mengendalikan semua penerimaan dan pengeluaran barang.
   c. Mendokumentasikan data-data PT.
   d. Melakukan pencatatan-pencatatan kegiatan PT.
4. Kepala Mekanik
   a. Mengatur serta membantu seluruh staff mekanik.
   b. Memberikan intruksi serta membantu para staff mekanik agar dapat memperbaiki bila terjadi kerusakan-kerusakan mesin di PT.

5. Pengawas Sortir

   Mengawasi dan mensortir setiap barang yang masuk maupun keluar.

6. Pengawas Giling Cuci

   Mengawasi semua kegiatan penggilingan cuci di PT.

7. Pengawas Pelet

   Mengawasi semua kegiatan pemeletan barang di PT.

8. Staf Mekanik

   Melakukan perbaikan apabila terjadi kerusakan-kerusakan pada mesin di PT.

2.2 Landasan Teori

Landasan teori merupakan teori-teori yang berhubungan dengan pembangunan sistem informasi di PT. Lastersia Plastisindo Karawang.

2.2.1 Sistem

Secara umum, sistem adalah suatu kesatuan dari dua atau lebih komponen yang berinteraksi untuk mencapai suatu tujuan. Pendekatan sistem yang menekankan pada komponen lebih mudah untuk dipelajari dan diterapkan pada berbagai bidang terutama untuk pengembangan sistem lebih lanjut. Pendefinisian sistem dengan penekanan elemen sebagai berikut :

> *"Sistem adalah kumpulan dari elemen-elemen yang berorientasi untuk mencapai suatu tujuan". (H.M Jogiyanto,1993:2)*

Bentuk umum sistem terdiri atas *input* (masukan), *output* (keluaran) dan proses. Seperti terlihat pada gambar berikut ini :

Gambar 2.2 Bentuk Umum Sistem

2.2.2 Informasi

Informasi merupakan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Nilai dari informasi ditentukan oleh dua hal yaitu manfaat dan biaya untuk mendapatkannya. Informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya untuk mendapatkannya. Untuk mendapatkan informasi yang berkualitas bergantung pada 3 hal, yaitu:

1. Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan, harus jelas mencerminkan maksudnya.
2. Tepat pada waktunya berarti informasi yang datang pada penerima tidak boleh terlambat, informasi yang sudah usang tidak akan mempunyai nilai lagi, karena informasi merupakan landasan didalam mengambil keputusan.
3. Relevan, berarti informasi tersebut mempunyai manfaat untuk memahami relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda.

Informasi merupakan hal yang sangat penting dalam mengambil keputusan, dan informasi tersebut dapat diperoleh dari suatu sistem informasi. Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi,

bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan bagi pihak luar tertentu berupa laporan-laporan yang diperlukan.

Dalam perkembangannya sistem informasi sangat erat hubungannya dengan teknologi komputer. Penggunaan teknologi komputer sangat diperlukan untuk membantu pengolahan data yang bersifat rutin dan membutuhkan ketelitian yang tinggi.

2.3 Perancangan Sistem

Perancangan sistem secara umum adalah suatu tahap dimana di dalamnya terdapat identifikasi komponen-komponen sistem informasi yang akan dirancang secara rinci yang bertujuan untuk memberikan gambaran kepada pengguna atau user mengenai sistem yang baru. Sedangkan desain sistem secara terinci dimaksudkan untuk pembuat program komputer dan ahli teknik lainnya yang akan mengimplementasikan sistem.

Penggambaran dan rancangan model sistem informasi secara logika dapat dibuat dalam bentuk Diagram Alir Data (DAD) atau *Data Flow Diagram* (DFD). Arus data pada Diagram Alir Data atau *Data Flow Diagram* dapat dijabarkan dengan menggunakan kamus data (*Data Dictionary*).

2.4 Model Proses

2.4.1 Diagram Aliran Data (*Data Flow Diagram*)

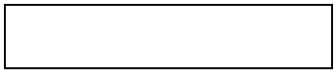
Diagram arus data merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur. Diagram arus data ini dapat menggambarkan sistem secara keseluruhan dan merupakan representasi dari sebuah sistem yang menunjukan proses dan aliran data yang ke luar ataupun ke dalam sistem. Definisi *Data Flow Diagram* adalah sebagai berikut :

"Data Flow Diagram adalah representasi grafik yang menggambarkan arus data dari suatu sistem. Data Flow Diagram menggambarkan komponen-komponen tersebut, asal dan tujuan serta penyimpanan data." (Lani Sidharta, Sistem Informasi Bisnis, 1995 : 65)

Simbol yang digunakan pada diagram alir data atau data flow diagram antara lain :

1. Kesatuan luar *(External Entity)*

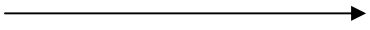
Kesatuan luar *(External Entity)* merupakan kesatuan di lingkungan luar sistem yang dapat berupa orang, organisasi, atau sistem lain. Lingkungan luar dan batas sistem dinotasikan dalam simbol yang sama.



Gambar 2.3 Notasi Kesatuan lingkungan

2. Arus Data (*Data Flow*)

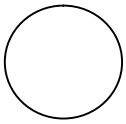
Arus data (*Data Flow*) di *DFD* diberi simbol suatu panah. Arus data ini mengalir diantara proses (*process*), simpanan data (*data store*) dan kesatuan luar (*external entity*).



Gambar 2.4 Notasi Arus Data

3. Proses (*Process*)

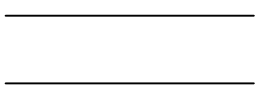
Proses adalah kegiatan yang dilakukan oleh manusia, mesin, atau komputer yang bertujuan mengolah data yang dimasukkan menjadi keluaran yang diinginkan.



Gambar 2.5 Notasi Proses

4. Simpanan Data (*File*)

Simpanan data disimbolkan dengan sepasang garis horizontal paralel. Simpanan data adalah tempat menyimpan atau untuk mengakses data yang diperlukan oleh sistem.



Gambar 2.6 Notasi Simpanan Data

Konsep dasar DFD dapat dibagi menjadi tiga bagian, yaitu :

a. Context Diagram (CD)

Merupakan diagram tingkat atas, yaitu diagram yang paling umum dari sebuah sistem informasi yang menggambarkan aliran-aliran data ke dalam dan ke luar sistem. Entitas-entitas eksternal adalah entitas-entitas yang terletak di luar sistem yang mengirim data atau menerima data dari sistem tersebut.

b. Middle Level

Merupakan pemecahan dari *context diagram* yang bisa disebut juga dengan DFD Level 0. DFD Level 0 ini menjelaskan proses yang lebih detail lagi daripada *context diagram* dan jika masih mungkin, bisa dipecah lagi menjadi level selanjutnya. Penomoran diagram dimulai dari 1,2,3 dan seterusnya.

c. Lowest Level

Merupakan pemecahan dari *middle level* yang masih mempunyai fungsi yang sama dengan level diatasnya. Penomoran dimulai dari 1.1, 1.2, 2.1, 2.2, dan seterusnya.

2.5 Kamus Data

Kamus data merupakan kumpulan data-data. Tujuannya adalah untuk memberikan informasi mengenai definisi struktur, pemakaian masing-masing elemen. Elemen adalah unit data yang terkecil. Elemen-elemen yang dapat menyusun sebuah sistem tersebut, terdiri dari :

1. Tujuan, merupakan tujuan dari sistem tersebut.
2. Batasan, merupakan batasan-batasan yang ada untuk mencapai tujuan dari sistem.
3. Kontrol, merupakan pengawas dari pelaksanaan pencapaian tujuan sistem.
4. Input, merupakan bagian dari sistem yang memberikan data masukan ke dalam sistem.
5. Proses, merupakan bagian yang memproses masukan data menjadi informasi yang sesuai dengan keinginan penerima.
6. Output, merupakan keluaran atau tujuan akhir dari sistem.
7. Umpan Balik, merupakan elemen sistem yang mempunyai tugas untuk melihat kembali apakah sistem telah berjalan sesuai dengan yang diinginkan.

Kamus data atau *Data dictionary* berisi keterangan tentang arus data, alias, bentuk data, penjelasan, periode, volume dan struktur data yang merupakan katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi.

2.6 Konsep Basis Data

Basis Data*(Database)* adalah kumpulan file-file yang saling berelasi, dan relasi tersebut biasa ditunjukkan dengan kunci dari tiap file yang ada. Satu database menunjukkan satu kumpulan data yang dipakai dalam satu lingkup organisasi. "*Database adalah kumpulan file-fle yang saling berelasi, relasi tersebut biasa ditunjukan dengan kunci dari tiap file yang ada.* (Ir. Harianto Kristianto, 1993:1).

Penggunaan database dalam suatu sistem mempunyai beberapa keuntungan, diantaranya:

a. Penggunaan data bersama

Data yang sama dapat diakses dalam waktu yang sama oleh beberapa user. Setiap aplikasi mempunyai file tersendiri sehingga berarti suatu data tunggal dalam basis data dapat digunakan untuk beberapa kegunaan.

b. Mengurangi redudansi data

Penyimpanan dibeberapa tempat untuk data yang sama disebut *redudansi.* Redudansi ini akan mengakibatkan pemborosan ruang penyimpanan dan biaya untuk mengakses data akan lebih tinggi. Penyimpanan data yang sama dengan kunci yang sama secara berulang-ulang pada beberapa file akan menyebabkan *inkonsistensi* data. Biasanya, kedua hal tadi terjadi ketika proses entry data.

c. Keamanan data dapat diaplikasikan

Database dapat mendefinisikan prosedur otoritas dan dapat mengatur pembagian akses untuk memastikan bahwa data hanya dapat diakses oleh user yang berhak.

d. Standarisasi

Database dapat melakukan kontrol terpusat untuk memastikan bahwa standard untuk nama data, pengguna data, format data atau dokumentasi diikuti seragam dalam suatu organisasi. Sehingga, keseragaman data dan informasi dapat disajikan untuk memenuhi kebutuhan user atau pengguna.

e. Integritas (Kesatuan)

Database berisi file yang saling berkaitan yaitu dengan adanya field kunci pada suatu file yang menghubungkan pada file lain, dengan tujuan untuk menjamin agar elemen dalam suatu file yang menunjuk ke suatu pengenal unik pada file lain benar-benar menunjuk pada suatu nilai yang memang ada.

2.7 Perancangan Basis Data

Perancangan basis data dapat dilakukan dengan dua cara, yaitu : Langsung membuat hubungan entitas (*Entity Relationship*). *Entity Relationship Diagram* atau dikenal dengan diagram E-R secara grafis menggambarkan isi sebuah database. Diagram ini memiliki dua komponen utama yaitu *Entity* dan relasi. Entity merupakan penjelasan dari orang, tempat, atau kejadian yang informasinya direkam. Relasi (*relationship*) adalah hubungan secara logis diantara entity-entity di dalam sistem informasi yang disajikan dalam bentuk ERD. Relasi antara dua tabel dapat dikategorikan menjadi tiga macam, yaitu :

a. Relasi satu ke satu (*One to One Relationship/1:1*)

   Hubungan antara tabel pertama dengan tabel kedua adalah satu berbanding satu, seperti pada pelajaran privat dimana satu guru mengajar satu siswa dan satu siswa hanya dapat belajar dengan satu guru.

b. Relasi satu ke banyak (*One to Many Relationship/1:n*)

Hubungan antara tabel pertama dengan tabel kedua adalah satu berbanding banyak atau sebaliknya. Seperti pada sistem pengajaran di sekolah dasar, dimana satu guru mengajar banyak siswa dan siswa hanya belajar dengan satu guru.

c. Relasi banyak ke banyak (*Many to Many Relationship/n:m*)

Hubungan antara tabel pertama dengan tabel kedua adalah banyak berbanding banyak. Seperti pada sistem pengajaran di perguruan tinggi, dimana satu dosen mengajar banyak mahasiswa, dan mahasiswa diajar oleh banyak dosen.

2.7.1 Komponen Data Model

a. Entity

Entity adalah sesuatu hal dalam bentuk apapun yang mana datanya dikumpulkan, dapat berupa objek orang, konsep abstrak, atau kejadian. Entity digambarkan dalam bentuk persegi panjang sebagai berikut :

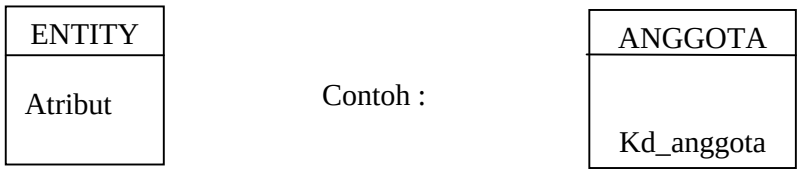


Gambar 2.7 Entity

Definisi Entity harus menekankan kepada pemakai tentang arti dari suatu entity, karena entity merupakan suatu abstrak, maka entity berlabel kata benda.

b. Atribut

Atribut adalah suatu karakteristik dari suatu entity. Berikut adalah simbol entity beserta atributnya.



Gambar 2.8 Atribut

Definisi suatu atribut termasuk informasi tentang :

1. Arti suatu atribut untuk pemakai
2. Karakteristik teknis dari suatu atribut bila atribut tersebut diimplementasikan sebagai bagian dari database untuk sistem.

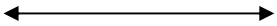
Definisi ini hendaknya dirancang mudah untuk dapat dimengerti oleh pemakai sistem.

c. Relationship

Relationship adalah hubungan atau asosiasi antara suatu entitas dengan dirinya sendiri atau dengan entitas lainnya. Suatu relationship bersifat binary.

Relationship digambarkan sebagai garis yang menghubungkan entitas-entitas yang dipandang memiliki hubungan antara satu dengan lainnya. Pada ujung masing-masing relationship terdapat derajat/cardinalitas hubungan. Terdapat dua macam relationship, yaitu *Mandatory* relationship dan *Optional* relationship. Mandatory digambarkan dengan garis penuh dan optional digambarkan dengan garis putus-putus.

(Mempergunakan diagram dari *Bachman*)

Hubungan satu ke satu 

Gambar 2.9 Hubungan satu ke satu

Hubungan satu ke banyak 

Gambar 2.10 Hubungan satu ke banyak

Hubungan banyak ke banyak 

Gambar 2.11 Hubungan banyak ke banyak

d. Identifier

Identifier adalah sebuah atau kumpulan atribut yang mempunyai nilai unik untuk setiap nilai yang terdapat pada atribut atau entitas.

2.7.2 Jaringan Client-Server

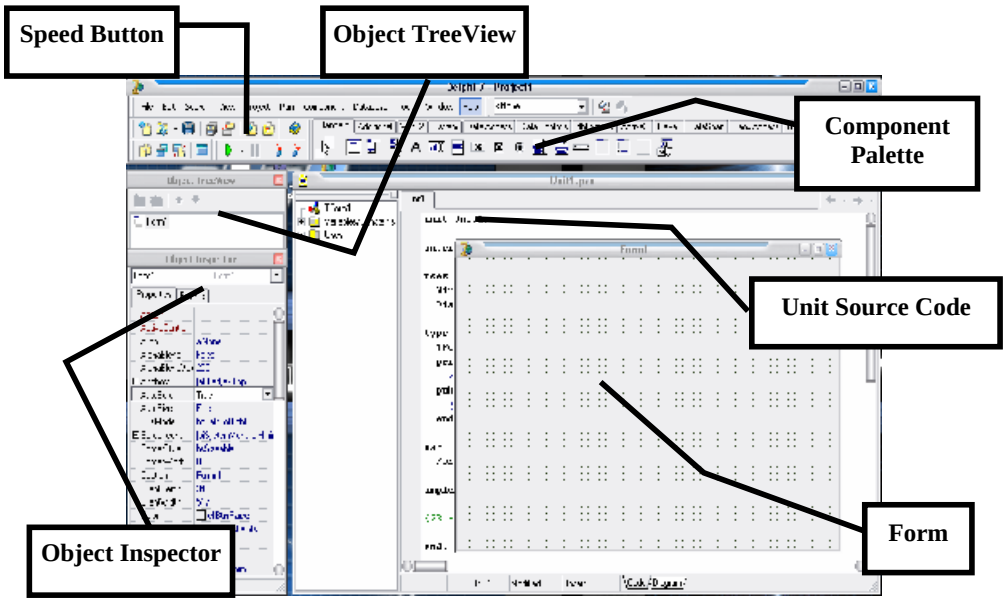
Sistem *Client Server* atau disebut juga sistem tersentralisasi diterapkan pada sebuah sistem jaringan. Sistem *Client Server* ini ditujukan untuk mengatasi kelemahan-kelemahan yang terdapat pada sistem sebelumnya. Sistem *Client Server* terdiri dari dua komponen utama yaitu *client* dan *server*. *Client* berisi aplikasi basis data dan *server* berisi DBMS dan basis data. Setiap aktivitas yang dikehendaki para pemakai akan lebih dulu ditangani oleh *client*. *Client* selanjutnya mengupayakan agar semua proses “sebisa mungkin” ditangani sendiri. Bila ada proses yang harus melibatkan data yang tersimpan pada basis data barulah *client* mengadakan hubungan dengan *server*.

Pada sistem *Client Server* untuk memenuhi kebutuhan, *client* akan mengirimkan *message* (perintah) *query* pengambilan data. Selanjutnya, *server* yang menerima

message tersebut akan menjalankan *query* tersebut dan hasilnya akan dikirimkan kembali ke *client*. Dengan begitu transfer datanya jauh lebih efisien.

2.8 Tinjauan perangkat lunak

Delphi adalah suatu program berbasis bahasa Pascal yang berjalan dalam lingkungan Windows. Delphi telah memanfaatkan suatu teknik pemrograman yang disebut RAD yang telah membuat pemrograman menjadi lebih mudah. Delphi adalah suatu bahasa pemrograman yang telah memanfaatkan metode pemrograman Object Oriented Programming (OOP).



Gambar 2.12 Tampilan Delphi 7

Berikut adalah keterangan dari gambar :

1. Speed Button : berisi tombol-tombol untuk mempercepat proses. Misal proses penyimpanan, membuka project baru, membuat form baru dan sebagainya.

2. Object Inspector : suatu window yang berguna untuk mengatur suatu object baik properti, events dan method.
3. Object TreeView : memperlihatkan list setiap object yang terdapat pada form dalam bentuk tree.
4. Form : Digunakan sebagai layar/window yang digunakan sebagai lembar kerja kita. Di form-lah semua komponen seperti tombol dan komponen lainnya disimpan.
5. Unit Source Code : Window/layar yang berisi perintah-perintah yang akan dieksekusi oleh komputer. Di layar inilah kita mengisikan program-program.
6. Component Palette : Layar yang berisikan komponen-komponen yang dipakai dipakai dalam program kita. Pallete tersebut meliputi Pallete Standart, Additional, Win32, System, Data Access, Data Control, ADO dan sebagainya.

2.9 Database MySQL

MySQL adalah multi *user database* yang menggunakan bahasa *Strucktured Query Language* (SQL). *MySQL* mampu menangani data yang cukup besar. Dengan menggunakan SQL, proses akses *database* menjadi lebih *user - friendly* dibandingkan dengan menggunakan *dBASE* atau *Clipper* yang masih menggunakan perintah - perintah pemrograman.

Perintah untuk mengelola *database* dibagi menjadi 3 kelompok, yaitu:

1. Perintah untuk mendefinisikan data/DDL (*Data Definition Language*).
2. Perintah untuk memanipulasi data/DML (*Data Manipulation Language*).

3. Perintah untuk mengendalikan data/DCL (*Data Control Language*).

2.9.1 DDL (*Data Definition Language*)

Merupakan Perintah untuk mendefinisikan data. Sebagai contoh, membuat *database*, membuat *table*, membuka *database*, menambah *field* baru, mengubah lebar *field*, menghapus *database*, menghapus *table*, membuat *index*, dan lainnya.

Adapun format penulisan perintahnya adalah sebagai berikut:

1. Membuat *database*.

    Create *Database* <nama*database*>;

2. Menghapus *database.*

    Drop Database <nama*database*>;

3. Membuat *table.*

    Create Table <namatabel>;

4. Menghapus *table.*

    Drop Table <namatabel>;

5. Membuat *file index.*

    Create Index <namaindex> ON <namatabel>

    (nama*field* 1[,nama*field* 2[,nama*field* 3..]]

6. Menghapus *file index.*

    Drop Index <namaindex> ON <namatabel>;

7. Menambah *field* baru.

    Alter Table<namatabel>;

    Add <namafieldbaru> tipe_data;

2.9.2 DML (*Data Manipulation Language*)

Merupakan perintah untuk memanipulasi data. Sebagai contoh perintah untuk memanipulasi data adalah memasukan *record* (*insert*), update data (*update*), menghapus *record* (*delete*), dan menampilkan *record* (*select*).

Adapun format penulisan perintahnya adalah sebagai berikut:

1. Perintah *insert*.

   Insert Into <namatabel> *values* (nilai1 [,nilai2[,…]]);

2. Perintah *update*.

   Update <namatabel> set *field*1=nilai1 [,*field*2=nilai2[,…]]

   [*where* kondisi];

3. Perintah *delete*.

   Delete from <namatabel> [*where* kondisi];

4. Perintah *select*.

   Select {* | *field*1 [,*field*2[,…]]} *from* namatabel [*where* kondisi];

2.9.3 DCL (*Data Control Language*)

Merupakan perintah untuk memberi hak otorisasi, untuk mengakses *database*. Perintah-perintah DCL, antara lain *Commit, Rollback, Grant* dan *Revoke.*

Adapun format penulisan perintahnya adalah sebagai berikut:

1. Menambah *user*.

   Insert into user (Host,User)

   Values (‘alamatkomputer’,’nama*user*’)

2. Mendefinisikan hak akses.

   Update User

Set namakolom1_Priv=’Y’

[,namakolom2_Priv=’Y’[,namakolom3_priv=’3’[,…]]

Where user =’nama*user*’

3. Mendefinisikan pemakai dan hak akses menggunakan DCL.

Grant [*privilege*][nama-namakolom]

ON nama*database*.namatabel

TO nama*user*@hostkomputer

IDENTIFIED BY “Password”

[WITH GRANT OPTION]

4. Mencabut hak akses.

Revoke [privilege] [nama-namakolom]

ON nama*database*.namatabel

FROM nama*user*@hostkomputer

5. Menghapus pemakai.

*Delete from User where User=’*nama*user*’

*Delete from Db where user=’*nama*user*’

*Delete from Tables_Priv where User=’*nama*user*’

Delete from Columns_Priv where User=’namauser’